
UNIVERSITI SAINS MALAYSIA

Second Semester Examination
2010/2011 Academic Session

April/May 2011

CPT113/CPM213 – Programming Methodology & Data Structures
[Metodologi Pengaturcaraan & Struktur Data]

Duration : 2 hours
[Masa : 2 jam]

INSTRUCTIONS TO CANDIDATE:
[ARAHAN KEPADA CALON:]

- Please ensure that this examination paper contains **TWO** sections in **ELEVEN** printed pages before you begin the examination.

*[Sila pastikan bahawa kertas peperiksaan ini mengandungi **DUA** bahagian di dalam **SEBELAS** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]*

- Answer **ALL** questions.

*[Jawab **SEMUA** soalan.]*

- You may answer the questions either in English or in Bahasa Malaysia.

[Anda dibenarkan menjawab soalan sama ada dalam bahasa Inggeris atau bahasa Malaysia.]

- In the event of any discrepancies, the English version shall be used.

[Sekiranya terdapat sebarang percanggahan pada soalan peperiksaan, versi bahasa Inggeris hendaklah diguna pakai.]

SECTION 1**MULTIPLE CHOICE QUESTIONS**

Select only **ONE (1)** correct answer. Each question carries 1 mark.

1. A node within a linked list can be implemented using a(n):
 - (A) Array
 - (B) Structure
 - (C) File
 - (D) Stack or Queue
 - (E) None of these

2. A primary feature of an ADT is:
 - (A) Information hiding
 - (B) Homogeneity
 - (C) Polymorphism
 - (D) None of these

3. The C++ operator used to *retrieve* data from a file is:
 - (A) *getline()*
 - (B) *cin*
 - (C) *>>*
 - (D) *<<*
 - (E) None of these

4. The concept of using several levels of addressing to access data is known as:
 - (A) inheritance
 - (B) polymorphism
 - (C) indirection
 - (D) overloading
 - (E) None of these

5. What type of inheritance occurs when a class member of a family can be traced back to several parent classes?
- (A) Single
 - (B) Virtual
 - (C) Polymorphic
 - (D) None of these
6. Overloaded functions are:
- (A) virtual
 - (B) statically bound
 - (C) dynamically bound
 - (D) polymorphic
 - (E) (A) and (C)
 - (F) (B) and (D)
 - (G) (B) and (C)
 - (H) None of these

(6/100)

TRUE or FALSE QUESTIONS

Indicate either **TRUE** or **FALSE** for the following questions. Each question carries 1 mark.

7. Since all of the designs for implementing stacks and queues use an array structure, we are free to access any element we wish in the random access array when working with a stack or queue ADT.
8. Before an element is removed from a stack or queue, you must first check to see if the stack or queue is empty.
9. Using the STL *list* class, you can insert or remove an element from both the front and the back of the list.
10. Classes are the real things that are manipulated by a program.

(4/100)

SECTION 2 – SHORT ANSWER

Answer **ALL** questions.

1. List and explain at least **two (2)** characteristics of an ADT. (4/100)
2. Explain, in general, one way to determine if a word is a palindrome using stacks and queues. (8/100)
3. Write the pseudocode steps required to insert a node at the front of a singly linked list. (8/100)
4. Write the pseudocode steps required to delete a node in the middle of a singly linked list. (4/100)
5. Why do you need a predecessor pointer when deleting a node from a linked list? (2/100)
6. Suppose that a dynamic linked list is used to implement a character stack such that `top` is an external pointer that locates the first node in the list, as well as the top of the stack.
 - (a) Write the pseudocode statements required to push an element onto the stack. (8/100)
 - (b) Write the pseudocode statement required to pop an element from the stack. (4/100)

For Questions 7 and 8, you may sketch a diagram or explain in words.

7. Suppose that `p` is pointing to some given node in a linked list. What is pointed to by the expression `next (next (p))`? (2/100)

8. What happens as a result of the following operation?

`Set next (p) = p`

(2/100)

9. Define a stack of strings called *myStringStack* using the STL *stack* class. (2/100)

10. Use the stack object in question 9 to push the string "C++" onto the stack. (2/100)

11. Write a loop that will empty and display the stack in question 10. (8/100)

12. Write the C++ code for an inorder binary tree traversal. (12/100)

13. (a) List down and explain the **three (3)** principles of object-oriented programming. (6/100)

- (b) By hand, trace through using the selection sort algorithm for the following list. Sort the list in increasing order. Count the number of comparisons made and number of times an entry moved.

27 32 36 30 21 15 24

(6/100)

14. Given the following recursive function:

```
long mystery(int n)
{
    if (n == 0)
        return 0;
    else
        return n+n * mystery(n-1);
}
```

(i) State the base case.

(3/100)

(ii) State the general case.

(3/100)

(iii) Find `mystery(4)`. Show your trace.

(6/100)

KERTAS SOALAN DALAM VERSI BAHASA MALAYSIA

[CPT113/CPM213]

- 7 -

BAHAGIAN 1

SOALAN PELBAGAI PILIHAN

Pilih **SATU (1)** jawapan betul sahaja. Satu soalan membawa 1 markah.

1. Suatu nod senarai berpaut boleh diimplementasikan dengan
 - (A) Tatasusunan
 - (B) Struktur
 - (C) Fail
 - (D) Stek atau Giliran
 - (E) Tiada di atas

2. Ciri utama ADT adalah:
 - (A) Penyembunyian maklumat
 - (B) Homogeniti
 - (C) Polimorfism
 - (D) Tiada di atas

3. Operator C++ yang diguna untuk mencapai data dari fail adalah
 - (A) *getline()*
 - (B) *cin*
 - (C) *>>*
 - (D) *<<*
 - (E) Tiada di atas

4. Konsep menggunakan beberapa lapisan pengalamatan untuk mencapai data dikenali sebagai:
- (A) warisan
 - (B) polimorfism
 - (C) ketakterusan
 - (D) penyaratan
 - (E) Tiada di atas
5. Apakah jenis warisan di mana ahli suatu kelas boleh disurih kembali ke beberapa kelas induk?
- (A) Tunggal
 - (B) Maya
 - (C) Polimorfik
 - (D) Tiada di atas
6. Fungsi lebih muatan adalah:
- (A) maya
 - (B) terikat secara statik
 - (C) terikat secara dinamik
 - (D) Polimorfik
 - (E) (A) and (C)
 - (F) (B) and (D)
 - (G) (B) and (C)
 - (H) Tiada di atas

(6/100)

SOALAN BENAR ATAU PALSU

Tanda **BENAR** atau **PALSU** bagi soalan-soalan berikut. Satu soalan membawa 1 markah.

7. Oleh kerana semua reka bentuk stek dan giliran menggunakan stuktur tatasusunan, kita boleh mencapai mana-mana elemen secara rawak dengan menggunakan tatasusunan pencapain rawak apabila berinteraksi dengan stek atau giliran.
8. Sebelum sesuatu elemen dihapuskan dari stek atau giliran, anda mesti menyamak samada stek atau giliran itu adalah kosong
9. Penggunaan kelas senarai STL, anda boleh menyelit dan menghapus elemen dari depan dan juga belakang sesuatu senarai.
10. Kelas adalah kod sebenar yang dimanipulasi oleh suatu atur cara.

(4/100)

BAHAGIAN 2 – JAWAPAN PENDEK

1. Senaraikan dan terangkan **dua (2)** ciri ADT.
2. Terangkan, secara umum, satu cara untuk mengenal pasti jika sesuatu perkataan adalah satu *palindrome* dengan menggunakan stek dan giliran.
3. Tulis langkah pseudokod yang diperlukan untuk menyelit satu nod di depan senarai berpaut tunggal
4. Tulis langkah pseudokod yang diperlukan untuk menghapus satu nod di tengah senarai berpaut tunggal.
5. Kenapa anda perlu guna penuding pendahulu semasa menghapus nod dari senarai berpaut?

(8/100)

(2/100)

(8/1000)

(4/100)

(2/100)

6. Andaikan suatu senarai berpaut dinamik diguna untuk mengimplimentasikan satu stek yang mana `top` adalah penuding luaran yang akan mencari nod pertama dalam senarai dan juga kedudukan atas stek.

(a) Tulis pseudokod untuk menolak satu elemen ke dalam stek.

(8/100)

(b) Tulis kenyataan pseudokod yang diperlukan untuk mengeluarkan elemen daripada suatu stek.

(4/100)

Untuk Soalan 7-8, anda boleh melakarkan satu rajah atau terangkan dengan perkataan:

7. Andaikan `p` menuding ke suatu nod dalam senarai berpaut. Apakah yang dituding oleh ungkapan `next(next(p))`?

(2/100)

8. Apa hasil operasi berikut?

`Set next(p) = p`

(2/100)

9. Isytihar satu stek rentetan yang dipanggil *myStringStack* dengan menggunakan kelas stek STL.

(3/100)

10. Guna objek stek dalam soalan 9 untuk menolak rentetan "C++" ke dalam stek.

(2/100)

11. Tulis satu gelung yang akan mengosongkan dan memapar stek dalam soalan 10.

(8/100)

12. Tulis kod C++ untuk pengelintaran pepohon dedua tak tersusun.

(12/100)

13. (a) Senarai dan terangkan **tiga (3)** prinsip pengaturcaraan berorientasi objek.

(6/100)

- (b) Surih dengan tangan, dengan menggunakan algoritma isih pilih untuk senarai berikut. Isih senarai mengikut turutan menaik. Kira jumlah perbandingan yang dibuat dan jumlah gerakan sesuatu kemasukan.

27 32 36 30 21 15 24

(6/100)

14. Diberikan fungsi rekursi berikut:

```
long mystery(int n)
{
    if (n == 0)
        return 0;
    else
        return n+n * mystery(n-1);
}
```

- (i) Nyatakan kes asas.

(3/100)

- (ii) Nyatakan kes am.

(3/100)

- (iii) Cari `mystery(4)`. Tunjukkan surihan anda.

(6/100)